IN THE CLAIMS:

Please amend claims 14 and 21-25 as follows.

1. (Original) A method for a programmable micro-controller comprising:

loading an instruction word within the micro-controller, the instruction word having a plurality of instruction fields; and

processing the plurality of instruction fields in parallel, each instruction field related to a specific operation for parsing a packet or encapsulating data to form a packet.

2. (Original) The method of claim 1, wherein the packet includes one or more protocol headers.

3. (Original) The method of claim 2, wherein the instruction word is loaded from a template, the template having a routine associated with each protocol header.

4. (Original) The method of claim 3, further comprising:

programming the micro-controller by adding a new routine to the template.

5. (Original) The method of claim 4, wherein the new routine is related to a new protocol.

6. (Original) The method of claim 1, wherein the processing of the plurality of instruction fields in parallel includes:

performing a memory load and store operation;

performing a checksum computation operation;

performing a test and compare operation;

performing a data extraction and insertion operation;

performing a branch target address operation; and

performing a branch trigger operation.

7. (Original) A programmable micro-controller comprising:

an embedded memory to store one or more instruction words, each instruction word including a plurality of instruction fields; and

one or more processing engines, each processing engine to process the plurality of instruction fields in parallel for each instruction word, each instruction field related to a specific operation for parsing a packet or encapsulating data to form a packet.

8. (Original) The programmable micro-controller of claim 7, wherein the packet includes one or more protocol headers.

9. (Original) The programmable micro-controller of claim 8, wherein the embedded memory stores a template, the template having a routine associated with each

protocol header.

10. (Original) The programmable micro-controller of claim 9, wherein the subsystem is programmed by adding a new routine to the template.

11. (Original) The programmable micro-controller of claim 10, wherein the new routine is related to a new protocol.

12. (Original) The programmable micro-controller of claim 7, wherein each processing engine includes:

an execution unit to perform a memory load and store operation;

an execution unit to perform a checksum computation operation;

an execution unit to perform a test and compare operation;

an execution unit to perform a data extraction and insertion operation;

an execution unit to perform a branch target address operation; and

an execution unit to perform a branch trigger operation.

13. (Original) The programmable micro-controller of claim 7, wherein the programmable micro-controller is a system on a chip.

14. (Currently amended) The programmable micro-controller of claim 7, further comprising:

an embedded buffer memory to store packets or data used in forming packets.

15. (Original) The programmable micro-controller of claim 7, wherein the instruction words are based on a Very Large Instruction Word (VLIW) architecture or on micro-code architecture.

16. (Original) A programmable micro-controller comprising:

an embedded buffer memory

a register set; and

programmable processing circuitry coupled to the embedded buffer memory and the register set, the programmable processing circuitry including a plurality of execution units, each execution unit to execute in parallel an operation within an instruction using the register set, the processing circuitry to parse a packet in the embedded buffer memory for extract data or to encapsulate data in the embedded buffer memory to form a packet using the execution units.

17. (Original) The programmable micro-controller of claim 16, wherein the register set includes a checksum register, buffer pointer register, micro-program register, branch program register, micro-instruction register, flags register, or a content table.

18. (Original) The programmable micro-controller of claim 16, wherein the operation includes a memory load and store operation, checksum operation, test and compare operation, data extraction and insertion operation, branch target address operation, or a branch trigger operation.

19. (Original) The programmable micro-controller of claim 18, wherein one of the execution units perform the memory load and store operation, checksum operation, test and compare operation, data extraction and insertion operation, branch target address operation, or the branch trigger operation.

20. (Original) The programmable micro-controller of claim 16, further comprising:

a template that is programmable, the template storing a plurality of routines, each routine associated with a different type of protocol.

21. (Currently amended) ~~An~~ A computer program embodied on computer-readable medium for controlling a micro-controller, wherein the instruction for ~~a~~ the micro-controller ~~comprising~~ comprises:

a plurality of operation fields to be processed in parallel by the micro-controller, each operation field related to a specific function for parsing a packet or encapsulating data to form a packet.

22. (Currently amended) The ~~instruction~~ computer program of claim 21, wherein, in the instruction, the operation fields include a memory load and store operation field, checksum operation field, test and compare operation field, data extraction and insertion operation field, branch target address operation field, or a branch trigger operation field.

23. (Currently amended) The ~~instruction~~ computer program of claim 21, wherein, in the instruction, each operation field is associated with a set of micro-instructions.

24. (Currently amended) The ~~instruction~~ computer program of claim 21, wherein the instruction is executed in a plurality of stages.

25. (Currently amended) The ~~instruction~~ computer program of claim 24, wherein, in the instruction, the stages include a prefetch stage, fetch stage, decode stage, and an execute stage.

26. (Original) A template within a system on a chip comprising:

a plurality of calls to routines, each routine associated with a particular protocol, each routine including one or more instructions, each instruction including a plurality of

operation fields that are processed in parallel to parse a packet or to encapsulate data to form a packet.

27. (Original) The template of claim 26, wherein the specific protocol includes existing protocols and new protocols.

28. (Original) The template of claim 26, wherein the template is programmable such that a new routine call can be added.

29. (Original) The template of claim 26, further comprising:

identifiers to identify routines for parsing a packet or encapsulating data to form a packet.

30. (Original) The template of claim 26, wherein the template is stored on an embedded memory.

31. (Original) In a router having a system on a chip, the system on a chip used to parse a packet or to encapsulate data to form a packet, a method for programming the system on a chip comprising:

downloading a routine for a new type of protocol to the system on a chip;

storing the downloaded routine in the system on a chip; and

adding a call to the stored routine in a template, the template tying routines together to parse a packet to extract data or to encapsulate data to form a packet.

32. (Original) The method of claim 31, wherein the downloading of the routine includes downloading the routine for a routing protocol.

33. (Original) The method of claim 31, wherein the downloading of the routine includes downloading the routine from a network or an external device.

34. (Original) The method of claim 33, wherein the downloading of the routine from a network includes downloading the routine from an Internet network.

35. (Original) The method of claim 31, wherein the adding of the call to the stored routine includes adding a call to the stored routine related to a new protocol.